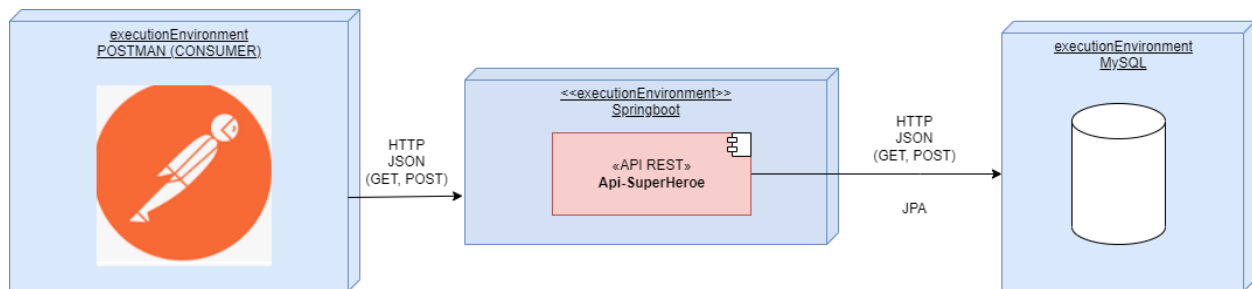


El proyecto que se encontrará en el repositorio fue desarrollado bajo los siguientes lineamientos:

- Springboot.
- Spring Security (JWT).
- Base de datos relacional MySQL.
- JPA.
- Pruebas unitarias a los métodos solicitados(en este caso únicamente las consultas y adicional a lo solicitado se agregó el poder insertar Mutantes en la base de datos).
- Manejo de excepciones a los métodos solicitados.

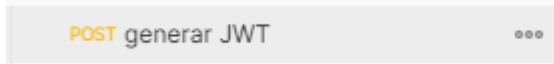
1. DIAGRAMA DE COMPONENTES



- ❖ El consumo del servicio se realizó por medio de postman ya que es una herramienta que facilita el diseño y desarrollo de sistemas backend.
- ❖ Se diseñó una api rest con springboot, con Json que permite el intercambio de información de forma sencilla
- ❖ La base de datos seleccionada fue MySQL, la cual es relacional ya que es más organizada en cuanto a registros y se basa más en la consistencia e integridad de los datos, mientras que una base de datos NoSQL se interesa más por la rapidez, siendo más utilizada para big data.

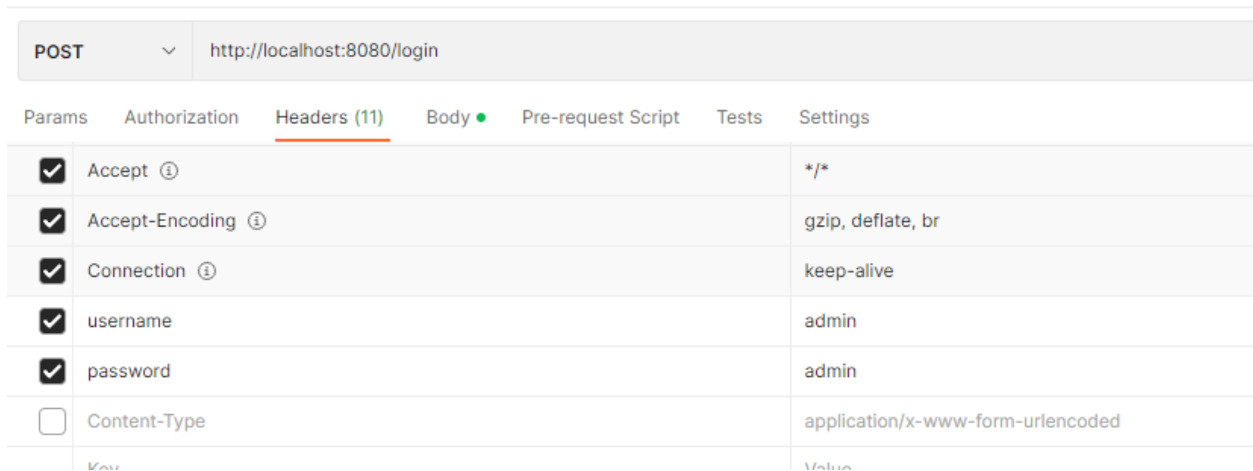
2. Instructivo consumo a través de postman

- El primer paso será generar el JWT con el siguiente request que encontrarán en la colección de postman

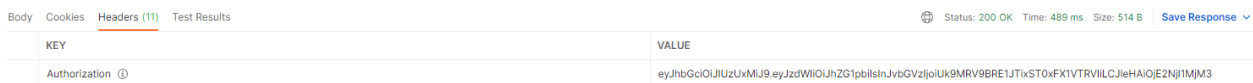


A través del siguiente endpoint: `http://localhost:8080/login`

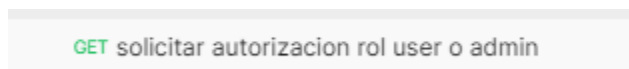
- Se deben configurar los siguientes headers:
 - Username con el valor de admin (se crearon dos roles admin y user)
 - Password con el valor de admin



- Entre los headers de respuesta se encuentra uno llamado **Authorization** que contiene el JWT generado, este se debe copiar para ser utilizado mas adelante en la autenticación



- Luego se debe acceder al siguiente método GET:



- A través del siguiente endpoint `http://localhost:8080/admin` para autenticarse con el JWT generado anteriormente

- El JWT se debe agregar como se indica en la imagen en la casilla de Token (Este proceso se debe realizar para todas las peticiones HTTP)

The screenshot shows the 'Authorization' tab of a REST client interface. At the top, the method is 'GET' and the URL is 'http://localhost:8080/admin'. Below the tabs (Params, Authorization, Headers (10), Body, Pre-request Script, Tests, Settings), the 'Type' is set to 'Bearer Token'. A warning message states: 'Heads up! These parameters hold sensitive data. To keep this data secure while working in a collat'. The 'Token' field contains the value 'eyJhbGciOiJIUzUxMiJ9.eyJzdWiiOiJhZG1pt...'. A note below the type says: 'The authorization header will be automatically generated when you send the request. [Learn more about authorization](#)'.

- Luego de haber realizado la autenticación tenemos la opción de insertar mutantes (ingresar nuevamente el JWT como en el paso anterior)

POST insertar mutantes

- Para acceder mediante el siguiente endpoint: `http://localhost:8080/personas/insertar`
- Podemos también consultar todos los mutantes alojados en la base de datos a través de la siguiente petición:

GET consultar mutantes

- A través del siguiente endpoint: `http://localhost:8080/mutante` (Se debe ingresar nuevamente el JWT) como se indica en la imagen

This screenshot is similar to the one above, showing the 'Authorization' tab for the endpoint 'http://localhost:8080/mutante'. The method is 'GET'. The 'Type' is 'Bearer Token' and the 'Token' field contains the same JWT value: 'eyJhbGciOiJIUzUxMiJ9.eyJzdWiiOiJhZG1pt...'. The same warning and note about automatic header generation are present.

- También podemos consultar mutantes por nombre con la siguiente petición:

GET consultar mutantes por nombre

- A través del siguiente endpoint: `http://localhost:8080/mutante/name?name=(aqui va el nombre del mutante)`

GET

▼

http://localhost:8080/mutante/name?name=Thor

Params ● Authorization ● Headers (7) Body Pre-request Script Tests Settings

Type

Bearer Token ▼

The authorization header will be automatically generated when you send the request.
[Learn more about authorization](#) ➔

! Heads up! These parameters hold sensitive data. To keep this data secure while working in a coll

Token

eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJhZG1pt...

- También podemos consultar mutantes por ubicación con la siguiente petición:

GET consultar mutantes por ubicacion

- A través del siguiente endpoint: `http://localhost:8080/mutante/location?location=Tierra`

GET

▼

http://localhost:8080/mutante/location?location=Tierra

Params ● Authorization ● Headers (7) Body Pre-request Script Tests Settings

Type

Bearer Token ▼

The authorization header will be automatically generated when you send the request.
[Learn more about authorization](#) ➔

! Heads up! These parameters hold sensitive data. To keep this data secure while working in a coll

Token

eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJhZG1pt...